实验报告12

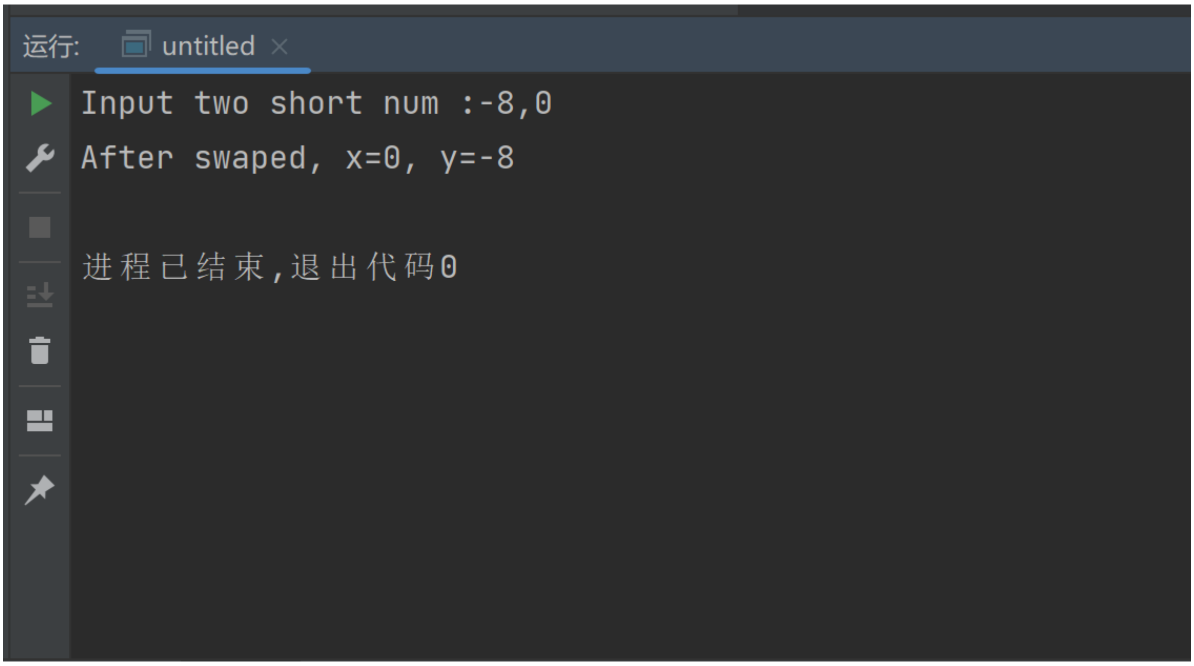
实验内容1

习题四（3）

修改后，代码如下：

```c
#include <stdio.h>
void swap(short *x, short *y) {
    short tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}
int main(void) {
    short x, y;
    printf("Input two short num :");
    scanf("%d", &x);
    scanf("%d", &y);
    if (x < y)
        swap(&x, &y);
    printf("After swaped, x=%d, y=%d\n", x, y);
}
```

运行结果如下



```
Input two short num :-8,0
After swaped, x=0, y=-8


进程已结束,退出代码0
```

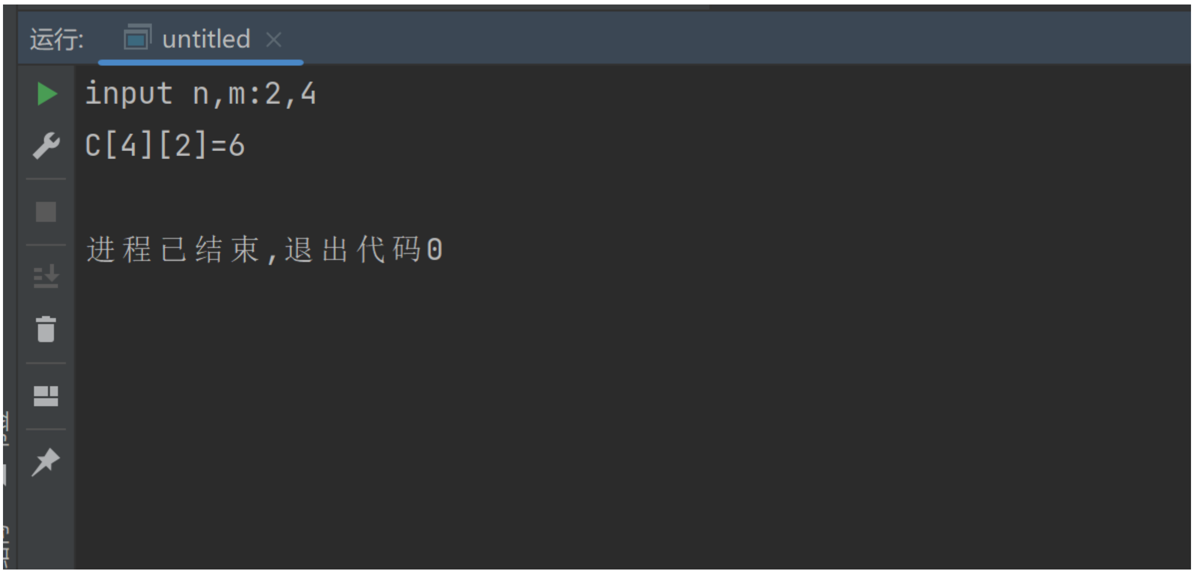
实验内容2

实验步骤（4）

代码编写如下

```
#include <stdio.h>
int combine(int n, int m) {
    if (n == 0)
        return 1;
    if (n == 1)
        return m;
    if (n > m / 2)
        return combine(m - n, m);
    return (combine(n, m - 1) + combine(n - 1, m - 1));
}
int main() {
    int n, m;
    printf("input n,m:");
    scanf("%d,%d", &n, &m);
    printf("C[%d][%d]=%d\n", m, n, combine(n, m));
}
```

运行结果：



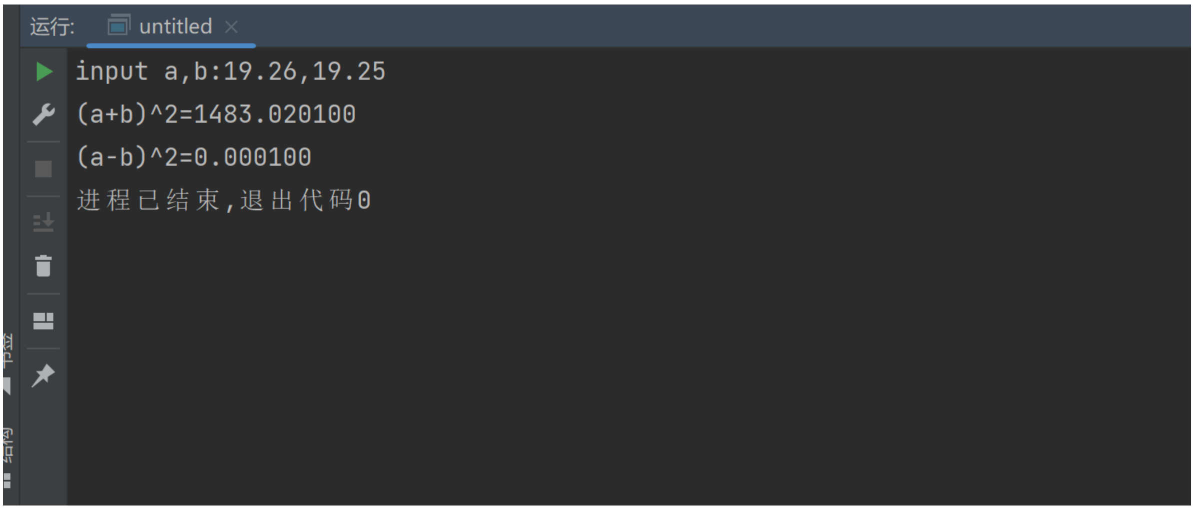
```
input n,m:2,4
C[4][2]=6

进程已结束,退出代码0
```

实验步骤（5）

代码编写如下：

```
#include <stdio.h>
double fun(double a, double b, double *result1, double *result2) {
    *result1 = (a + b) * (a + b);
    *result2 = (a - b) * (a - b);
}
int main() {
    double a, b, result1, result2;
    printf("input a,b:");
    scanf("%lf,%lf", &a, &b);
    fun(a, b, &result1, &result2);
    printf("(a+b)^2=%lf\n(a-b)^2=%lf", result1, result2);
}
```

运行结果：



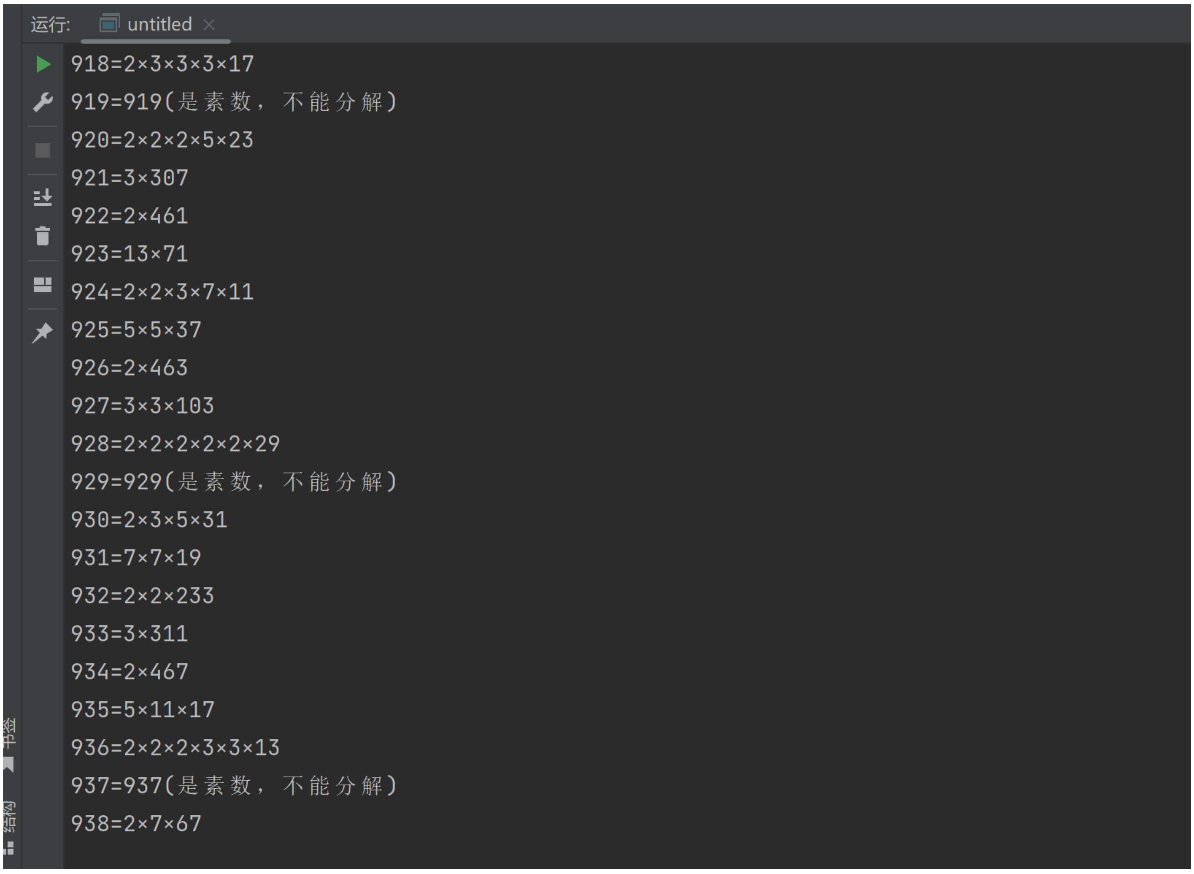
```
input a,b:19.26,19.25
(a+b)^2=1483.020100
(a-b)^2=0.000100
进程已结束,退出代码0
```

实验步骤（6）

代码编写如下

```
#include <stdio.h>
void FindPrime(int *prime) {
    prime[0] = 2;
    int count = 1, i, num;
    for (num = 3; num <= 1000; num++) {
        for (i = 2; i < num; i++) {
            if (num % i == 0)
                break;
            if (i == num - 1)
                prime[count++] = num;
        }
    }
}
short exist(int n, int *prime) {
    for (int i = 0; i < 168; i++) {
        if (prime[i] == n)
            return 1;
        else if (prime[i] > n || i == 167)
            return 0;
    }
}
void fun(int num, int *prime) {
    printf("\n%d=", num);
    if (exist(num, prime))
        printf("%d(是素数，不能分解)", num);
    else {
        int i = 0;
        for (i = 0; i < 168; i++) {
            if (num % prime[i] == 0) {
                printf("%d", prime[i]);
                num /= prime[i];
                i = -1;
                if (num != 1) {
                    printf("x");
                }
            }
        }
    }
}
int main() {
    int prime[168];
    FindPrime(prime);
    for (int num = 100; num < 1000; num++)
        fun(num, prime);
}
```

部分运行结果如下:



```
918=2×3×3×3×17
919=919(是素数，不能分解)
920=2×2×2×5×23
921=3×307
922=2×461
923=13×71
924=2×2×3×7×11
925=5×5×37
926=2×463
927=3×3×103
928=2×2×2×2×2×29
929=929(是素数，不能分解)
930=2×3×5×31
931=7×7×19
932=2×2×233
933=3×311
934=2×467
935=5×11×17
936=2×2×2×3×3×13
937=937(是素数，不能分解)
938=2×7×67
```